

[illegible]

```
FFFFFFFFF 000000 RRRRRRRR VV VV MM MM
FFFFFFFFF 000000 RRRRRRRR VV VV MM MM
FF 00 00 RR RR VV VV MMMM MMMM
FF 00 00 RR RR VV VV MMMM MMMM
FF 00 00 RR RR VV VV MM MM
FF 00 00 RR RR VV VV MM MM
FFFFFFFFF 00 00 RRRRRRRR VV VV MM MM
FFFFFFFFF 00 00 RRRRRRRR VV VV MM MM
FF 00 00 RR RR VV VV MM MM
FF 00 00 RR RR VV VV MM MM
FF 00 00 RR RR VV VV MM MM
FF 00 00 RR RR VV VV MM MM
FF 000000 RR RR VV VV MM MM
FF 000000 RR RR VV VV MM MM
```

```
....
....
....
....
```

```
LL IIIIII SSSSSSSS
LL IIIIII SSSSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LL II SSSSSS
LL II SSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS
```



```

1 0001 0 MODULE FOR$$VM ( ! Internal FORTRAM Virtual memory allocation/deallocation
2 0002 0 IDENT = '1-001' ! File: FORVM.B32
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1
7 0007 1 *****
8 0008 1 *
9 0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
10 0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
11 0011 1 * ALL RIGHTS RESERVED.
12 0012 1 *
13 0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
14 0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
15 0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
16 0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
17 0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
18 0018 1 * TRANSFERRED.
19 0019 1 *
20 0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
21 0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
22 0022 1 * CORPORATION.
23 0023 1 *
24 0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
25 0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
26 0026 1 *
27 0027 1 *
28 0028 1 *****
29 0029 1
30 0030 1 ++
31 0031 1 FACILITY:FORTRAN support library
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 Dynamic virtual memory allocation and deallocation.
36 0036 1 FORTRAN interface with LIB$GET_VM and LIB$FREE_VM
37 0037 1 resource allocation procedures.
38 0038 1
39 0039 1 ENVIRONMENT: User access mode; mixture of AST level or not.
40 0040 1 This module is both shared and non-shared. Hence all
41 0041 1 EXTERNAL references are of type GENERAL to prevent data truncation errors
42 0042 1 when linking with the non-shared FORTRAN compatibility routines.
43 0043 1
44 0044 1 AUTHOR: T. Hastings, CREATION DATE: 4-Dec-77; Version 01
45 0045 1
46 0046 1 MODIFIED BY:
47 0047 1
48 0048 1 01 - original
49 0049 1 0-2 - Use FOR$$SIG_FATINT. TNH 5-Dec-77
50 0050 1 0-3 - Don't clear memory. TNH 8-Dec-77
51 0051 1 0-04 - Change REQUIRE files for VAX system build. DGP 28-Apr-78
52 0052 1 0-05 - Add optional second arg (FCB only). TNH 22-MAY-78
53 0053 1 0-06 - Use FOR$$SIG_DATCOR instead of FOR$$SIG_FATINT. TNH 10-June-78
54 0054 1 0-07 - Make all external references GENERAL, since this module
55 0055 1 - is both shared and non-shared. TNH 3-Aug-78
56 0056 1 0-08 - Change file name to FORVM.B32, and change the names of
57 0057 1 the REQUIRE files similarly. JBS 14-NOV-78

```

FOR\$VM
1-001

¹₅
16-Sep-1984 00:57:30
14-Sep-1984 12:33:00

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[FORRTL.SRC]FORVM.B32;1 Page (1) 2

: 58
: 59

0058 1 ! 1-001 - Update version number and copyright notice. JBS 16-NOV-78
0059 1 !--


```

61      0060 1  |
62      0061 1  | TABLE OF CONTENTS:
63      0062 1  |
64      0063 1  |
65      0064 1  | FORWARD ROUTINE
66      0065 1  |     FOR$$GET_VM,      ! Allocate virtual memory - interface
67      0066 1  |     FOR$$FREE_VM: NOVALUE; ! Deallocate virtual memory - interface
68      0067 1  |
69      0068 1  |
70      0069 1  |
71      0070 1  | INCLUDE FILES:
72      0071 1  |
73      0072 1  |
74      0073 1  |     REQUIRE 'RTLML:FORERR';      ! FORTRAN error numbers
75      0141 1  |     REQUIRE 'RTLIN:RTLPSECT';    ! Define DECLARE_PSECTS macro
76      0236 1  |
77      0237 1  |
78      0238 1  | MACROS:
79      0239 1  |
80      0240 1  |     NONE
81      0241 1  |
82      0242 1  | EQUATED SYMBOLS:
83      0243 1  |
84      0244 1  |
85      0245 1  |
86      0246 1  |
87      0247 1  | PSECT DECLARATIONS:
88      0248 1  |
89      0249 1  |
90      0250 1  |     DECLARE_PSECTS (FOR);      ! declare PSECTs for FOR$ facility
91      0251 1  |
92      0252 1  |
93      0253 1  | OWN STORAGE:
94      0254 1  |
95      0255 1  |
96      0256 1  | EXTERNAL REFERENCES:
97      0257 1  |
98      0258 1  | EXTERNAL ROUTINE
99      0259 1  |
100     0260 1  | +
101     0261 1  | MAINTENANCE NOTE: Since this module is called by FORTRAN compatibility
102     0262 1  | routines which are un-shared and the entry points are not vectored,
103     0263 1  | a separate copy of this module is linked with the user program when
104     0264 1  | the user calls a FORTRAN compatibility routine. In order to prevent
105     0265 1  | data truncation errors from the linker, all external references are
106     0266 1  | of addressing mode general (rather than word displacement) even for
107     0267 1  | the same PSECT.
108     0268 1  | -
109     0269 1  |
110     0270 1  |     FOR$$SIGNAL STO: ADDRESSING_MODE (GENERAL) NOVALUE,      ! FORTRAN SIGNAL_STOP for current unit
111     0271 1  |     FOR$$SIG_DATCOR: ADDRESSING_MODE (GENERAL) NOVALUE,      ! FORTRAN SIGNAL_STOP OTSS_INTDATCOR
112     0272 1  |                                                                ! INTERNAL DATA CORRUPTED IN RUN-TIME LIBRAR
113     0273 1  |     FOR$$SIG_NO_LUB: ADDRESSING_MODE (GENERAL) NOVALUE,      ! FORTRAN SIGNAL_STOP when no current LUB/IS
114     0274 1  |     LIB$GET_VM: ADDRESSING_MODE (GENERAL); ! LIBRARY allocate virtual memory
115     0275 1  |     LIB$FREE_VM: ADDRESSING_MODE (GENERAL); ! LIBRARY deallocate virtual memory
116     0276 1  |

```

```
118 0277 1 GLOBAL ROUTINE FOR$$GET_VM (      ! Allocate dynamic virtual memory
119 0278 1      NUM_BYTES,                      ! longword size in bytes
120 0279 1      LOGICAL_UNIT)                ! optional logical unit (if LUB/ISB/RAB not allocated)
121 0280 1      =
122 0281 1
123 0282 1      ++
124 0283 1      FUNCTIONAL DESCRIPTION:
125 0284 1
126 0285 1          Allocates n virtually contiguous bytes at an arbitrary place in
127 0286 1          the program region and returns the virtual address of the first byte.
128 0287 1          See description of library LIB$GET_VM for details.
129 0288 1          This procedure is provided only for convenience to FORTRAN support library.
130 0289 1          It checks for errors and SIGNAL_STOPs any.
131 0290 1          It does not clear core for speed.
132 0291 1
133 0292 1      CALLING SEQUENCE:
134 0293 1
135 0294 1          ALLOC_ADR.wa.v = FOR$$GET_VM (NUM_BYTES.rlu.v [, logical_unit.rlu.v])
136 0295 1
137 0296 1      INPUT PARAMETERS:
138 0297 1
139 0298 1          num_bytes is an unsigned longword integer value
140 0299 1          specifying the number of virtually contiguous bytes to
141 0300 1          be allocated. Sufficient pages are allocated to
142 0301 1          satisfy the request. However, the program should not
143 0302 1          reference before the first byte address assigned
144 0303 1          (base_address) beyond the last byte assigned
145 0304 1          (base_adr+num_bytes - 1) since it may be assigned to
146 0305 1          another procedure.
147 0306 1
148 0307 1          [logical_unit.rlu.v] Optional logical unit number. Used only if
149 0308 1          an error occurs and LUB/ISB/RAB is not already allocated.
150 0309 1
151 0310 1      OUTPUT PARAMETERS:
152 0311 1
153 0312 1          None.
154 0313 1
155 0314 1      IMPLICIT INPUTS:
156 0315 1
157 0316 1          OTSS$A_CUR_LUB contains the address of the current LUB/ISB/RAB
158 0317 1          for which any errors detected will be signaled.
159 0318 1          See also LIB$GET_VM.
160 0319 1
161 0320 1      IMPLICIT OUTPUTS:
162 0321 1
163 0322 1          See LIB$GET_VM.
164 0323 1
165 0324 1      FUNCTION VALUE:
166 0325 1
167 0326 1          The address of the block allocated is returned
168 0327 1          as the function value.
169 0328 1
170 0329 1      SIDE EFFECTS:
171 0330 1
172 0331 1          The following errors are SIGNAL_STOPped:
173 0332 1
174 0333 1          FOR$_INSVIRMEM indicates 'INSUFFICIENT VIRTUAL MEMORY' when the
```



```

: 175      0334 1  program
: 176      0335 1  region was attempted to be expanded.
: 177      0336 1  OTSS_INTDATCOR indicates 'BAD BLOCK SIZE either 0 oor
: 178      0337 1  larger than FOR$K_MXVMBLK.
: 179      0338 1  No partial assignment is made.
: 180      0339 1  An appropriate number of virtual bytes are removed from the image
: 181      0340 1  free memory list. If needed the program region is expanded by
: 182      0341 1  calling the SYS$EXPREG system service. if too large a size is
: 183      0342 1  requested or the program region could not be expended as needed.
: 184      0343 1  --
: 185      0344 1
: 186      0345 2  BEGIN
: 187      0346 2  BUILTIN
: 188      0347 2  ACTUALCOUNT;
: 189      0348 2  LOCAL
: 190      0349 2  TEMP_ADR; ! Adr. of block allocated
: 191      0350 2  IF NOT LIB$GET_VM (NUM_BYTES, TEMP_ADR)
: 192      0351 2  THEN
: 193      0352 3  BEGIN
: 194      0353 3  IF ACTUALCOUNT() GTRU 1
: 195      0354 3  THEN
: 196      0355 3  FOR$$SIG_NO_LUB (FOR$K_INSVIRMEM, .LOGICAL_UNIT)
: 197      0356 3  ELSE
: 198      0357 3  FOR$$SIGNAL_STO (FOR$K_INSVIRMEM)
: 199      0358 2  END;
: 200      0359 2  RETURN .TEMP_ADR;
: 201      0360 1  END; ! end of FOR$GET_VM routine
```

```

                                .TITLE  FOR$$VM
                                .IDENT  \1-001\

                                .EXTRN  FOR$$SIGNAL_STO
                                .EXTRN  FOR$$SIG_DATCOR
                                .EXTRN  FOR$$SIG_NO_LUB
                                .EXTRN  LIB$GET_VM, LIB$FREE_VM

                                .PSECT  _FOR$CODE, NOWRT, SHR, PIC, 2

                                .ENTRY  FOR$$GET_VM, Save nothing
                                SUBL2   #4, SP
                                PUSHL   SP
                                PUSHAB  NUM_BYTES
                                CALLS   #2, LIB$GET_VM
                                BLBS    R0, 2$
                                CMPB    (AP), #1
                                BLEQU   1$
                                PUSHL   LOGICAL_UNIT
                                PUSHL   #41
                                CALLS   #2, FOR$$SIG_NO_LUB
                                BRB     2$
                                PUSHL   #41
                                CALLS   #1, FOR$$SIGNAL_STO
                                MOVL    TEMP_ADR, R0
                                RET

                                0000 00000
                                5E      04  C2 00002
                                SE      DD 00005
                                04      AC 9F 00007
                                00000000G 00 02 FB 0000A
                                1C      50 E8 00011
                                01      6C 91 00014
                                08      0E 1B 00017
                                AC      DD 00019
                                00000000G 00 29 DD 0001C
                                02      FB 0001E
                                09      11 00025
                                29      DD 00027 1$:
                                00000000G 00 01 FB 00029
                                50      6E D0 00030 2$:
                                04      00033
```

```

                                .ENTRY  FOR$$GET_VM, Save nothing
                                SUBL2   #4, SP
                                PUSHL   SP
                                PUSHAB  NUM_BYTES
                                CALLS   #2, LIB$GET_VM
                                BLBS    R0, 2$
                                CMPB    (AP), #1
                                BLEQU   1$
                                PUSHL   LOGICAL_UNIT
                                PUSHL   #41
                                CALLS   #2, FOR$$SIG_NO_LUB
                                BRB     2$
                                PUSHL   #41
                                CALLS   #1, FOR$$SIGNAL_STO
                                MOVL    TEMP_ADR, R0
                                RET

                                : 0277
                                : 0350
                                : 0353
                                : 0355
                                : 0357
                                : 0359
                                : 0360
```

; Routine Size: 52 bytes, Routine Base: _FOR\$CODE + 0000

FORSSVM
1-001

M 5
16-Sep-1984 00:57:30
14-Sep-1984 12:33:00

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[FORRTL.SRC]FORVM.B32;1 Page 6 (3)


```
203 0361 1 GLOBAL ROUTINE FOR$$FREE_VM (      ! Internal FORTRAN deallocate virtual memory
204 0362 1      NUM_BYTES,                      ! size in bytes
205 0363 1      BASE_ADR)                      ! ADR. of block to be deallocated
206 0364 1      : NOVALUE                      =
207 0365 1
208 0366 1  ++
209 0367 1  FUNCTIONAL DESCRIPTION:
210 0368 1
211 0369 1      Deallocates n virtually contiguous bytes starting at the
212 0370 1      specified virtual address. The number of bytes actually
213 0371 1      allocated is rounded up so that the smallest number of whole quad
214 0372 1      words are de-allocated. Numerous error checks are made to make
215 0373 1      sure that the block being returned is a legitimate free area.
216 0374 1
217 0375 1  CALLING SEQUENCE:
218 0376 1
219 0377 1      CALL FOR$$FREE_VM(num_bytes.rlu.v, base_adr.ra.v)
220 0378 1
221 0379 1  INPUT PARAMETERS:
222 0380 1
223 0381 1      num_bytes is an unsigned integer
224 0382 1      specifying the number of virtually contiguous bytes to
225 0383 1      be deallocated.
226 0384 1
227 0385 1      base_adr is the address of
228 0386 1      the first byte to be deallocated.
229 0387 1
230 0388 1  OUTPUT PARAMETERS:
231 0389 1
232 0390 1  None.
233 0391 1
234 0392 1  IMPLICIT INPUTS
235 0393 1
236 0394 1      OTSS$A_CUR_LUB contains the address of the current LUB/ISB/RAB
237 0395 1      for which the storage is being returned. Any errors
238 0396 1      are signaled on the logical unit.
239 0397 1
240 0398 1  IMPLICIT OUTPUTS
241 0399 1
242 0400 1      The pages are deallocated by calling $DEALTVA. Then the pages
243 0401 1      are marked as available in the OWN storage maintained by
244 0402 1      LIB$GET_VM.
245 0403 1
246 0404 1  COMPLETION STATUS:
247 0405 1
248 0406 1      None.
249 0407 1
250 0408 1  SIDE EFFECTS:
251 0409 1
252 0410 1      Any errors are signal_stopped on the current logical unit.
253 0411 1      OTSS_INTDATCOR indicates BAD BLOCK ADDRESS
254 0412 1
255 0413 1      PUTS the indicated block back on the image free storage list.
256 0414 1  --
257 0415 1
258 0416 2  BEGIN
259 0417 2
```


FOR\$\$VM
1-001

B 6
16-Sep-1984 00:57:30
14-Sep-1984 12:33:00

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[FORRTL.SRC]FORVM.B32;1 Page 8 (4)

```
: 260      0418  2      |+
: 261      0419  2      | Deallocate virtual memory, SIGNAL_STOP OTSS_INTDATCOR if error
: 262      0420  2      | -
: 263      0421  2
: 264      0422  2      IF NOT LIB$FREE_VM (NUM_BYTES, BASE_ADR) THEN FOR$$SIG_DATCOR ();
: 265      0423  1      END;
```

```
                                0000 00000
                                08 AC 9F 00002
                                04 AC 9F 00005
                                00000000G 00 02 FB 00008
                                           07 50 E8 0000F
                                00000000G 00 00 FB 00012
                                           04 00019 1$:
                                .ENTRY FOR$$FREE_VM, Save nothing
                                PUSHAB BASE_ADR
                                PUSHAB NUM_BYTES
                                CALLS #2, LIB$FREE_VM
                                BLBS R0, 1$
                                CALLS #0, FOR$$SIG_DATCOR
                                RET
                                : 0361
                                : 0422
                                :
                                :
                                : 0423
```

; Routine Size: 26 bytes, Routine Base: _FOR\$CODE + 0034

```
: 266      0424  1 END
: 267      0425  0 ELUDOM
```

PSECT SUMMARY

Name	Bytes	Attributes
_FOR\$CODE	78	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

COMMAND QUALIFIERS

; BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS\$:FORVM/OBJ=OBJ\$:FORVM MSRC\$:FORVM/UPDATE=(ENH\$:FORVM)

```
: Size:      78 code + 0 data bytes
: Run Time:   00:03.7
: Elapsed Time: 00:12.9
: Lines/CPU Min: 6967
: Lexemes/CPU-Min: 17508
: Memory Used: 32 pages
: Compilation Complete
```


0185

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY